



日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application:

2000年11月15日

出 願 番 号  
Application Number:

特願2000-348475

出 願 人  
Applicant(s):

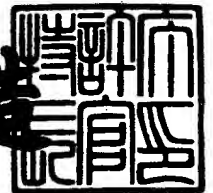
富士通株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 2月23日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2001-3011268

【書類名】 特許願

【整理番号】 0001048

【提出日】 平成12年11月15日

【あて先】 特許庁長官 及川 耕造 殿

【国際特許分類】 G06F 9/28

【発明の名称】 情報処理装置

【請求項の数】 7

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 安藤 寿茂

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100070150

【住所又は居所】 東京都渋谷区恵比寿4丁目20番3号 恵比寿ガーデンプレイスタワー32階

【弁理士】

【氏名又は名称】 伊東 忠彦

【電話番号】 03-5424-2511

【先の出願に基づく優先権主張】

【出願番号】 平成11年特許願第371617号

【出願日】 平成11年12月27日

【手数料の表示】

【予納台帳番号】 002989

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704678

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置

【特許請求の範囲】

【請求項 1】 種々の処理のうちで特定の処理をそれ以外の処理よりも頻繁に実行する情報処理装置であって、

該種々の処理に対応する命令セットを実行可能な第 1 のプロセッサと、

該命令セットの一部或いは全てを実行可能であり該特定の処理に対応する該命令セットの一部を該第 1 のプロセッサより効率的に実行可能な第 2 のプロセッサを含み、該特定の処理を該第 2 のプロセッサが実行し該特定の処理以外の処理を該第 1 のプロセッサが実行することを特徴とする情報処理装置。

【請求項 2】 全ての処理は最初に該第 2 のプロセッサに割り当てられ、該特定の処理に対応する該命令セットの一部以外の命令を実行必要な場合に、割り込みにより該第 2 のプロセッサから第 1 のプロセッサに処理を移すことを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 全ての処理は最初に該第 2 のプロセッサに割り当てられ、該第 2 のプロセッサで実行不可能或いは効率的な実行が出来ない命令が出現した場合に、割り込みにより該第 2 のプロセッサから第 1 のプロセッサに処理を移すことを特徴とする請求項 1 記載の情報処理装置。

【請求項 4】 該実行不可能或いは効率的な実行が出来ない命令は、浮動小数点演算命令であることを特徴とする請求項 3 記載の情報処理装置。

【請求項 5】 該第 2 のプロセッサは、マルチスレッド方式及びマルチプロセッサ方式のうちの少なくとも 1 つの方式により処理を並列に実行することで、該特定の処理に対応する該命令セットの一部を該第 1 のプロセッサより効率的に実行可能であることを特徴とする請求項 1 記載の情報処理装置。

【請求項 6】 該第 1 のプロセッサは汎用プロセッサであり、該第 2 のプロセッサは該特定の処理であるトランザクション処理を効率的に実行可能なように構成されたトランザクション処理プロセッサであることを特徴とする請求項 1 記載の情報処理装置。

【請求項 7】 該第 1 のプロセッサ及び該第 2 のプロセッサがメモリ空間を共

有することを特徴とする請求項 1 記載の情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は一般に情報処理装置に関し、詳しくは汎用処理及びトランザクション処理を行なう情報処理装置に関する。

【0002】

【従来の技術】

現在の汎用高性能プロセッサは、マルチメディア処理等を含む幅広い用途に対して高い性能を発揮することが要求されるため、様々な処理に適した種々の命令を高速に実行出来るように設計されている。例えば、浮動小数点演算部を設けて、浮動少数点演算が必要な処理を高速に実行できるように構成されている。また汎用高性能プロセッサは、通常、一連の命令からなる 1 つのプログラムを高速に実行することに重点をおいて設計されている。例えば、分岐方向が確定する前に分岐命令を実行するために分岐予測をしたり、オペランド待ちのためのリザベーションを行ったり、オペランドの処理順序変更のためにリオーダ処理を実行する機能等が設けられている。

【0003】

しかしながら、近年計算機に対する需要が特に延びているのは、トランザクション処理のシステムやウェブサーバシステム等の分野である。トランザクション処理やウェブサーバ処理で実行対象となる処理では、論理演算・整数演算が中心であり、浮動小数点演算等の複雑な処理は殆ど必要とされない。またトランザクション処理やウェブサーバ処理においては、1 つのプログラムの処理を高速に実行することよりも、小規模の処理を大量かつ並列に実行する機能が重要になる。

現在の汎用高性能プロセッサは、上述のようにオールラウンドな高性能を重視して設計されているため、単純なトランザクション処理やウェブサーバ処理を実行させるためには余分な機能が多すぎる。またトランザクション処理やウェブサーバ処理では、単純な処理の大量並列実行が必要であるのに、1 つのプログラムの処理を高速に実行することを重視した汎用高性能プロセッサでは、そのような

要求に充分に応えることが出来ない。従って、トランザクション処理やウェブサーバ処理等を高速に実行できる情報処理装置が望まれている。

## 【 0 0 0 4 】

図 1 に高性能汎用プロセッサの一例のブロック構成図を示す。

## 【 0 0 0 5 】

高性能汎用プロセッサ 1 は、命令キャッシュ&命令フェッチ 2、分岐予測部 3、プログラムカウンタ 4、チェックポイント部 5、固定小数点レジスタ 6、浮動小数点レジスタ 7、固定小数点リオーダバッファ 8、浮動小数点リオーダバッファ 9、固定小数点リザベーション部 10、浮動小数点リザベーション部 11、ロード／ストアリザベーション部 12、固定小数点演算部 13、浮動小数点演算部 14、ロード／ストア部 15、データキャッシュ 16 から構成される。

## 【 0 0 0 6 】

命令キャッシュ&命令フェッチ 2 は、命令を保持すると共に、実行する命令をフェッチする。フェッチした命令がデコードされ、デコード結果に従って、高性能汎用プロセッサ 1 の各部が制御される。分岐予測部 3 は、デコードされた命令の中から条件付分岐命令を検出して、分岐条件が確定する前に分岐方向を予測する。プログラムカウンタ 4 は、順次実行される命令のアドレスを指し示す。チェックポイント部 5 は、分岐予測に従って分岐命令を実行したときに、レジスタ値等を含むプロセッサステータスを格納する。分岐予測がはずれたときには、チェックポイント部 5 に格納された情報を読み出して、レジスタ値等を含むプロセッサステータスを分岐前の状態に戻すことで、分岐ミスから回復する。

## 【 0 0 0 7 】

固定小数点レジスタ 6 は、固定小数点算術論理演算に対するオペランドを格納し、命令実行に必要なオペランドを固定小数点リザベーション部 10 に供給すると共に、ロード／ストア命令に対するオペランドを格納し、命令実行に必要なオペランドをロード／ストアリザベーション部 12 に供給する。固定小数点リオーダバッファ 8 は、固定小数点演算部 13 によって計算された計算結果を格納すると共に、ロード／ストア部 15 によって実行されたロード命令の結果を格納する。固定小数点リオーダバッファ 8 は、要求される最新のオペランドが固定小数点

レジスタ 6 ではなく固定小数点リオーダバッファ 8 にあることを検出すると、そのオペランドを固定小数点リザベーション部 1 0 或いはロード／ストアリザベーション部 1 2 に直接に供給する。また固定小数点リオーダバッファ 8 は、命令実行結果を固定小数点レジスタ 6 に供給することで、オペランドの処理順序が変更された場合であっても、所定のレジスタへオペランドを格納して命令実行を終了する。

## 【 0 0 0 8 】

浮動小数点レジスタ 7 は、浮動小数点算術演算に対するオペランドを格納し、命令実行に必要なオペランドを浮動小数点リザベーション部 1 1 に供給すると共に、ロード／ストア命令に対するオペランドを格納し、命令実行に必要なオペランドをロード／ストアリザベーション部 1 2 に供給する。浮動小数点リオーダバッファ 9 は、浮動小数点演算部 1 4 によって計算された計算結果を格納すると共に、ロード／ストア部 1 5 によって実行されたロード命令の結果を格納する。浮動小数点リオーダバッファ 9 は、要求される最新のオペランドが浮動小数点レジスタ 7 ではなく浮動小数点リオーダバッファ 9 にあることを検出すると、そのオペランドを浮動小数点リザベーション部 1 1 或いはロード／ストアリザベーション部 1 2 に直接に供給する。また浮動小数点リオーダバッファ 9 は、命令実行結果を浮動小数点レジスタ 7 に供給することで、オペランドの処理順序が変更された場合であっても、所定のレジスタへオペランドを格納して命令実行を終了する。

## 【 0 0 0 9 】

固定小数点リザベーション部 1 0 は、実行する固定小数点命令を保持し、必要なオペランドが固定小数点レジスタ 6 或いは固定小数点リオーダバッファ 8 から供給されるのを待つ。必要な全てのオペランドが得られると、その命令は実行可能状態となる。固定小数点リザベーション部 1 0 は、実行可能状態の命令を選択して、固定小数点演算部 1 3 に供給する。固定小数点演算部 1 3 に同時に供給可能な命令の最大数は、固定小数点演算部 1 3 に設けられる固定小数点演算ユニットの数に等しい。

## 【 0 0 1 0 】

浮動小数点リザベーション部 1 1 は、実行する浮動小数点命令を保持し、必要なオペランドが浮動小数点レジスタ 7 或いは浮動小数点リオーダバッファ 9 から供給されるのを待つ。必要な全てのオペランドが得られると、その命令は実行可能状態となる。浮動小数点リザベーション部 1 1 は、実行可能状態の命令を選択して、浮動小数点演算部 1 4 に供給する。浮動小数点演算部 1 4 に同時に供給可能な命令の最大数は、浮動小数点演算部 1 4 に設けられる浮動小数点演算ユニットの数に等しい。

## 【 0 0 1 1 】

ロード／ストアリザベーション部 1 2 は、実行するロード／ストア命令を保持し、必要なオペランドが固定小数点レジスタ 6、浮動小数点レジスタ 7、固定小数点リオーダバッファ 8、或いは浮動小数点リオーダバッファ 9 から供給されるのを待つ。必要な全てのオペランドが得られると、その命令は実行可能状態となる。ロード／ストアリザベーション部 1 2 は、実行可能状態の命令を選択して、ロード／ストア部 1 5 に供給する。ロード／ストア部 1 5 に同時に供給可能な命令の最大数は、ロード／ストア部 1 5 に設けられるロード／ストア実行ユニットの数に等しい。

## 【 0 0 1 2 】

固定小数点演算部 1 3 は、固定小数点リザベーション部 1 0 から供給された命令を実行し、命令実行結果を固定小数点リオーダバッファ 8 に送る。浮動小数点演算部 1 4 は、浮動小数点リザベーション部 1 1 から供給された命令を実行し、命令実行結果を浮動小数点リオーダバッファ 9 に送る。またロード／ストア部 1 5 は、ロード／ストアリザベーション部 1 2 から供給されたロード／ストア命令を実行して、その結果を固定小数点リオーダバッファ 8 或いは浮動小数点リオーダバッファ 9 に供給する。

## 【 0 0 1 3 】

ストア命令を実行すると、ロード／ストア部 1 5 はデータをデータキャッシュ 1 6 に書き込み、ロード命令を実行すると、ロード／ストア部 1 5 はデータをデータキャッシュ 1 6 から読み出す。

## 【 0 0 1 4 】



高性能汎用プロセッサ 1 は、固定小数点演算部 1 3 及び浮動小数点演算部 1 4 を有し、論理／整数演算及び浮動小数点演算の両方に対応可能とされている。また、分岐予測部 3、固定小数点リオーダーバッファ 8、浮動小数点リオーダーバッファ 9、固定小数点リザベーション部 1 0、浮動小数点リザベーション部 1 1、及びロード／ストアリザベーション部 1 2 が設けられており、分岐方向が確定する前に分岐命令を実行するために分岐予測をしたり、オペランド待ちのためのリザベーションを行ったり、オペランドの処理順序変更のためにリオーダー処理を実行することが可能となっている。

## 【0 0 1 5】

このように汎用高性能プロセッサ 1 では、オールラウンドな高性能を重視して種々の命令を高速に実行出来るように設計されていると共に、一連の命令からなる 1 つのプログラムを高速に実行することに重点をおいて設計されている。

## 【0 0 1 6】

なお上記の高性能汎用プロセッサ 1 では、固定小数点演算部 1 3、浮動小数点演算部 1 4、及びロード／ストア部 1 5 が各々複数ユニットずつ設けられており、ある程度の並列処理が可能である。しかし並列に処理できる最大命令数は、一般に 6 ～ 8 命令程度であり、大量の処理を並列に実行するように設計されているのではない。

## 【0 0 1 7】

図 2 はマルチスプレッドトランザクション処理システムのブロック構成図を示す。

## 【0 0 1 8】

マルチスプレッド方式のトランザクション処理システム 1 7 は、 $n$  個のトランザクション処理プロセッサコア 1 8 - 1 ～ 1 8 -  $n$ 、メモリ 1 9、入出力インタフェース 2 0、及びシステムバス 2 1 から構成される。

## 【0 0 1 9】

トランザクション処理プロセッサコア 1 8 - 1 ～ 1 8 -  $n$  の各々は、マルチスプレッド処理のためにプログラムカウンタを複数個備え、トランザクション処理を効率よく実行できる構成とされている。メモリ 1 9 は、システムバス 2 1 を介し

て  $n$  個のトランザクション処理プロセッサコア 18-1 ~ 18- $n$  と接続されており、 $n$  個のトランザクション処理プロセッサコア 18-1 ~ 18- $n$  によって共有される。

#### 【0020】

入出力インタフェース 20 は、外部とシステムバス 21 とのインタフェースを提供する。

#### 【0021】

図 3 はトランザクション処理プロセッサコアの一例のブロック構成図を示す。同図中、図 1 と同一の構成部分には同一の参照番号を付し、その説明は省略する。

#### 【0022】

トランザクション処理プロセッサコア 18-1 ~ 18- $n$  には、トランザクション処理で殆ど使用されない浮動小数点演算器は設けられていない。また 1 つのプログラムを高速に実行するよりは複数のプログラムを並列に大量に実行することに重点がおかれているので、分岐予測、リオーダ、及びリザベーション等を実行するための機能は設けられておらず、マルチスレッド化のためにプログラムカウンタ 23 及びレジスタ 24 が複数個設けられている。例えば、ATM のトランザクション処理であれば、あるプログラムカウンタによりユーザ A の引き落とし処理を実行し、別のプログラムカウンタによりユーザ B の引き落とし処理を実行するということにより、複数の処理を並行して実行することが可能になる。

#### 【0023】

このような構成とすることにより、ある処理に対応する所定の命令列でメモリの待ち時間が発生したとき、他の処理に対応する命令列を処理できるので、命令の実行効率を向上することができる。このようにして、図 3 に示すトランザクション処理プロセッサは、トランザクション処理を効率よく処理することが出来る。

#### 【0024】

図 2 に示すマルチスレッドトランザクション処理システム 15 は、図 3 に示すようなトランザクション処理プロセッサを  $n$  個備えることにより、小規模な処

理を大量に並列実行することを可能とし、更に効率よくトランザクション処理を実行出来る構成となっている。なお図 3 に示すトランザクション処理プロセッサは回路規模が比較的小さいので、数多くのプロセッサを同一のチップ上に搭載することが可能である。

【 0 0 2 5 】

【発明が解決しようとする課題】

図 1 に示す高性能汎用プロセッサ 1 は、オールラウンドな高性能を重視して設計されているため、単純なトランザクション処理やウェブサーバ処理を実行させるためには余分な機能が多すぎる。またトランザクション処理やウェブサーバ処理では、単純な処理の大量並列実行が必要であるのに、1 つのプログラムの処理を高速に実行することを重視した汎用高性能プロセッサでは、そのような要求に充分に応えることが出来ない。仮に複数のプロセッサを用意しようとしても、高性能汎用プロセッサは回路規模が大きいために、1 チップに搭載できるプロセッサ数に限界があり、充分に並列度を上げることが出来ない。

【 0 0 2 6 】

また図 2 に示すマルチスプレッドトランザクション処理システム 1 5 だけでウェブサーバシステム等を構成することは、トランザクション処理の効率化は達成できても、科学技術演算等の浮動小数点演算を必要とする複雑な処理に高速に対応できなくなると共に、単一のプログラムを高速に実行する種々の機能を持たないので、単一で大規模な処理或いは複雑な処理が要求されたときには、効率が著しく低下することになる。

【 0 0 2 7 】

本願発明は上記の点に鑑みてなされたもので、多様な命令を効率よく処理できる情報処理装置を提供することを目的とする。

【 0 0 2 8 】

【課題を解決するための手段】

本発明によれば、種々の処理のうちで特定の処理をそれ以外の処理よりも頻繁に実行する情報処理装置は、種々の処理に対応する命令セットを実行可能な第 1 のプロセッサと、該命令セットの一部或いは全てを実行可能であり特定の処理に

対応する該命令セットの一部を第 1 のプロセッサより効率的に実行可能な第 2 のプロセッサを含み、特定の処理を第 2 のプロセッサが実行し特定の処理以外の処理を第 1 のプロセッサが実行することを特徴とする。

## 【 0 0 2 9 】

上記発明では、種々の処理を実行可能な第 1 のプロセッサと、種々の処理のうち特定の処理を第 1 のプロセッサよりも効率的に実行可能な第 2 のプロセッサとを設けて、特定の処理を第 2 のプロセッサに実行させ特定の処理以外の処理を第 1 のプロセッサに実行させることで、全体として効率的な処理を実現することが出来る。

## 【 0 0 3 0 】

本発明のある側面によれば、具体的には、第 1 のプロセッサは汎用プロセッサであり、第 2 のプロセッサは特定の処理であるトランザクション処理を効率的に実行可能なように構成されたトランザクション処理プロセッサである。またトランザクション処理プロセッサは、マルチスレッド方式及びマルチプロセッサ方式のうちの少なくとも 1 つの方式により処理を並列に実行することで、特定の処理を汎用プロセッサよりも効率的に実行可能なように構成されている。

## 【 0 0 3 1 】

上記発明では、オールラウンドな高性能を重視して設計されると共に、1 つのプログラムの処理を高速に実行することを重視して設計された汎用高性能プロセッサと、小規模な処理を大量に並列実行することを重視して設計されるマルチスレッド或いはマルチプロセストランザクション処理プロセッサとを設け、トランザクション処理はトランザクション処理プロセッサに割り当てて、トランザクション処理プロセッサが効率的に実行できない処理は汎用高性能プロセッサに割り当ててすることで、多様な命令を効率よく処理できる情報処理装置を提供することが出来る。

## 【 0 0 3 2 】

また本発明のある側面によれば、上記発明において、全ての処理は最初に第 2 のプロセッサに割り当てられ、特定の処理に対応する命令セットの一部以外の命令を実行必要な場合に、割り込みにより第 2 のプロセッサから第 1 のプロセッサ

に処理を移すことを特徴とする。

【 0 0 3 3 】

上記発明では、トランザクション処理プロセッサで、例えば浮動小数点演算を実行する命令が発生した場合、汎用プロセッサに割り込みをかけて、汎用プロセッサで浮動小数点演算処理以降の処理を実行する。トランザクション処理プロセッサから汎用プロセッサにプロセス処理を引き渡す契機となる命令としては、浮動小数点演算命令以外にも、トランザクション処理では効率が悪いが汎用プロセッサでは効率的に実行可能な命令であればよい。

【 0 0 3 4 】

【発明の実施の形態】

以下に本発明の実施例を、添付の図面を用いて詳細に説明する。

【 0 0 3 5 】

図 4 は本発明の第 1 の実施例によるシステム構成図を示す。

【 0 0 3 6 】

本実施例の情報処理装置 3 1 は、汎用プロセッサ 3 2 - 1、3 2 - 2、トランザクション処理プロセッサ 3 3 - 1、3 3 - 2、メインメモリ 3 4、入出力インタフェース 3 5、システムコントローラ 3 6、及びシステムバス 3 7 を含む。

【 0 0 3 7 】

汎用プロセッサ 3 2 - 1 及び 3 2 - 2 は、図 1 に示す汎用プロセッサ 1 と同様な構成である。即ち、整数演算及び論理演算並びに浮動小数点演算が実行可能とされていると共に、分岐方向が確定する前に分岐命令を実行するために分岐予測をしたり、オペランド待ちのためのリザベーションを行ったり、オペランドの処理順序変更のためにリオーダ処理を実行すること等が可能となっている。なお汎用プロセッサ 3 2 - 1 及び 3 2 - 2 の数は 2 つに限られるものではなく、任意の数設けられてよい。

【 0 0 3 8 】

トランザクション処理プロセッサ 3 3 - 1 及び 3 3 - 2 は、整数演算及び論理演算以外の複雑な演算機能や制御機能は含んでいなくてよく、マルチスレッド化及び／又はマルチプロセッサ化されており、トランザクション処理が効率的に実

行可能な構成とされる。なおトランザクション処理プロセッサ 3 3 - 1 及び 3 3 - 2 の数は 2 つに限られるものではなく、任意の数設けられてよい。

#### 【 0 0 3 9 】

メインメモリ 3 4 は、R A M (Random Access Memory) から構成され、データを記憶する。メインメモリ 3 4 は、システムバス 3 7 を介して汎用プロセッサ 3 2 - 1、3 2 - 2 及びトランザクション処理プロセッサ 3 3 - 1、3 3 - 2 に接続される。メインメモリ 3 4 に記憶されたデータは、汎用プロセッサ 3 2 - 1、3 2 - 2 及びトランザクション処理プロセッサ 3 3 - 1、3 3 - 2 で共有される。

#### 【 0 0 4 0 】

入出力インタフェース 3 5 は、システムバス 3 7 と外部周辺装置とのインタフェースを提供する。システムコントローラ 3 6 は、汎用プロセッサ 3 2 - 1、3 2 - 2、トランザクション処理プロセッサ 3 3 - 1、3 3 - 2 からのアクセス要求に応じて、メインメモリ 3 4 及びシステムバス 3 7 を制御する。

#### 【 0 0 4 1 】

図 5 は、トランザクション処理プロセッサ 3 3 - 1 の構成を示す構成図である。

#### 【 0 0 4 2 】

図 5 のトランザクション処理プロセッサ 3 3 - 1 は、トランザクション処理演算部 4 1 及び 2 次キャッシュメモリ 4 2 を含む。

#### 【 0 0 4 3 】

トランザクション処理演算部 4 1 は 1 つの半導体チップから構成され、2 次キャッシュメモリ 4 2 が外付けされる。2 次キャッシュメモリ 4 2 は、R A M から構成され、データ及び命令を一時的に記憶する。

#### 【 0 0 4 4 】

トランザクション処理演算部 4 1 は、m 個のプロセッサコア 4 3 - 1 乃至 4 3 - m、2 次キャッシュタグ 4 4、インタフェース 4 5、及び内部バス 4 6 を含む。

#### 【 0 0 4 5 】

プロセッサコア 4 3 - 1 乃至 4 3 - m は、図 3 に示すトランザクション処理プロセッサコアと同様な構成を有するものでよい。即ち、プロセッサコア 4 3 - 1 乃至 4 3 - m の各々はマルチスレッド化されて、トランザクション処理を効率よく処理できるように構成される。また浮動小数点演算等の複雑な演算機能は設けられていなくてよく、更に分岐方向が確定する前に分岐命令を実行するために分岐予測をしたり、オペランド待ちのためのリザベーションを行ったり、オペランドの処理順序変更のためにリオーダ処理を実行する等の機能も不在でよい。

【 0 0 4 6 】

なおプロセッサコア 4 3 - 1 乃至 4 3 - m が複数個提供されることで、トランザクション処理の並列実行は可能となるので、各プロセッサコアがマルチスレッド化されていることは必須要件ではない。また極端な場合には、プロセッサコア 4 3 - 1 が 1 つだけ設けられている場合であっても、このプロセッサコアがトランザクション処理に特化されていればよい。

【 0 0 4 7 】

プロセッサコア 4 3 - 1 乃至 4 3 - m は、内部バス 4 6 を介して、2 次キャッシュタグ 4 4 及びインタフェース 4 5 に接続されている。2 次キャッシュタグ 4 4 は、2 次キャッシュメモリ 4 2 の使用状況を管理する。インタフェース 4 5 は、2 次キャッシュメモリ 4 2 と内部バス 4 6 とのインタフェースを提供する。

【 0 0 4 8 】

以下に、本実施例のトランザクション処理プロセッサ 3 3 - 1 及び 3 3 - 2 が実行する処理について説明する。

【 0 0 4 9 】

図 6 は本発明の第 1 の実施例によるトランザクション処理プロセッサの処理フローチャートを示す。

【 0 0 5 0 】

トランザクション処理プロセッサ 3 3 - 1 は、ステップ S 1 - 1 乃至 S 1 - 5 を実行する。

【 0 0 5 1 】

ステップ S 1 - 1 は、命令を発行する処理である。ステップ S 1 - 1 で命令が

発行されると、次にステップ S 1 - 2 が実行される。

【 0 0 5 2 】

ステップ S 1 - 2 は、命令が固定小数点演算か否かを判定する処理である。発行された命令が固定小数点演算の場合には、ステップ S 1 - 3 が実行される。

【 0 0 5 3 】

ステップ S 1 - 3 は、命令を実行する処理である。ステップ S 1 - 3 で命令が実行されると、次にステップ S 1 - 4 が実行される。

【 0 0 5 4 】

ステップ S 1 - 4 は、次の命令の有無を判定する処理である。ステップ S 1 - 4 で、次に命令があると判定されると、ステップ S 1 - 1 に戻って次の命令が発行される。また、ステップ S 1 - 4 で次の命令がないと判定されると、処理を終了する。

【 0 0 5 5 】

また、ステップ S 1 - 2 で命令が浮動小数点演算であると判定された場合には、次にステップ S 1 - 5 が実行される。

【 0 0 5 6 】

ステップ S 1 - 5 は、割り込み処理である。ステップ S 1 - 5 の割り込み処理については後述する。ステップ S 1 - 5 で、割り込み処理が終了すると処理を終了する。

【 0 0 5 7 】

以下に、ステップ S 1 - 5 の割り込み処理について詳細に説明する。

【 0 0 5 8 】

図 7 は本発明の第 1 の実施例によるトランザクション処理プロセッサの割り込み処理の処理フローチャートを示す。

【 0 0 5 9 】

トランザクション処理プロセッサ 3 3 - 1 は、ステップ S 1 - 5 の割り込み処理としてステップ S 2 - 1 ~ S 2 - 6 を実行する。

【 0 0 6 0 】

ステップ S 2 - 1 は、発生された割り込み処理を解析する。ステップ S 2 - 1



で割り込み処理を解析すると、次にステップ S 2 - 2 を実行する。

【 0 0 6 1 】

ステップ S 2 - 2 は、ステップ S 2 - 1 での解析結果が通常の割り込み処理か否かを判定する処理である。ステップ S 2 - 2 で、通常の割り込み処理であると判定すると、次に、ステップ S 2 - 3 が実行される。ステップ S 2 - 3 は、通常の割り込み処理を実行する処理である。

【 0 0 6 2 】

また、ステップ S 2 - 2 で、浮動小数点演算を実行するための割り込み処理であると判定された場合には、ステップ S 2 - 4 が実行される。

【 0 0 6 3 】

ステップ S 2 - 4 は、実行の引継ぎに必要なトランザクション処理プロセッサの内部状態をメモリに格納し、命令アドレスと状態を格納したメモリアドレスとを内部レジスタに格納する処理である。ステップ S 2 - 4 で、内部レジスタに命令と状態メモリのアドレスが格納されると、次にステップ S 2 - 5 が実行される。

【 0 0 6 4 】

ステップ S 2 - 5 は、汎用プロセッサ 3 2 - 1 又は 3 2 - 2 に割り込みをかける処理である。この割り込みは、割り込みと共に上記内部レジスタの内容を含む割り込みベクトルを、トランザクション処理プロセッサから汎用プロセッサ 3 2 - 1 又は 3 2 - 2 に供給することで行われる。

【 0 0 6 5 】

以上により、トランザクション処理プロセッサ 3 3 - 1、3 3 - 2 で浮動小数点演算を実行する命令が発生した場合に、汎用プロセッサ 3 2 - 1 又は 3 2 - 2 に割り込みがかけられる。

【 0 0 6 6 】

次に、汎用プロセッサ 3 2 - 1 の処理について説明する。

【 0 0 6 7 】

図 8 は本発明の第 1 の実施例による汎用プロセッサの処理フローチャートを示す。

## 【 0 0 6 8 】

汎用プロセッサ 3 2 - 1 は、以下のステップ S 3 - 1 乃至 S 3 - 6 に示す処理を実行する。

## 【 0 0 6 9 】

ステップ S 3 - 1 は、他のプロセッサからの割り込み要求の受信の有無を判定する。ステップ S 3 - 1 で、他のプロセッサからの割り込み要求がなければ、ステップ S 3 - 2 が実行され、処理を終了する。ステップ S 3 - 2 は、通常の処理を実行する処理である。

## 【 0 0 7 0 】

また、ステップ S 3 - 1 で、他のプロセッサから割り込み要求があれば、次にステップ S 3 - 3 が実行される。

## 【 0 0 7 1 】

ステップ S 3 - 3 は、割り込み要求が命令の実行依頼か否かを判定する処理である。ステップ S 3 - 3 で、割り込み要求が命令の実行依頼でなければ、次にステップ S 3 - 4 が実行され、処理が終了する。ステップ S 3 - 4 は、通常の割り込み処理を実行する処理である。

## 【 0 0 7 2 】

また、ステップ S 3 - 3 で、割り込み要求が命令の実行依頼であると判定されると、次にステップ S 3 - 5 が実行される。

## 【 0 0 7 3 】

ステップ S 3 - 5 は、割り込み要求と共にトランザクション処理プロセッサ 3 2 - 1 から供給された割り込みベクトルを受け取り、割り込みベクトルに含まれる状態メモリアドレスを用いて汎用プロセッサの内部に状態を複製し、更にこの割り込みベクトルから命令アドレスを取り出す処理である。次にステップ S 3 - 6 が実行される。ステップ S 3 - 6 は、上記命令アドレスの命令、即ち実行依頼があった命令から実行を開始する処理である。

## 【 0 0 7 4 】

以上により汎用プロセッサ 3 2 - 1、3 2 - 2 で浮動小数点演算とそれ以降の処理が実行される。

## 【 0 0 7 5 】

以上のように本実施例によれば、トランザクション処理プロセッサ 3 2 - 1、3 2 - 2 で、浮動小数点演算を実行する命令が発生した場合、汎用プロセッサ 3 1 - 1 に割り込みをかけて、汎用プロセッサ 3 1 - 1、3 1 - 2 で浮動小数点演算処理を実行する。ここで浮動小数点演算は、トランザクション処理プロセッサから汎用プロセッサにプロセス処理を引き渡す契機としての一例であり、このような契機としては浮動小数点演算以外にも、トランザクション処理では効率が悪いが汎用プロセッサでは効率的に実行可能な命令であればよく、本発明はそのような命令実行を契機とする形態一般を含むものである。

## 【 0 0 7 6 】

本発明においては、上記のような構成とすることで、トランザクション処理を主たる処理として実行しながらも汎用処理に対応したシステムにおいて、トランザクション処理を効率的に実行することが可能となる。なお OS（オペレーティングシステム）は、逐次実行比率が高いので、汎用プロセッサ 3 1 - 1 又は 3 1 - 2 のいずれかで実行するようにする。但しこれは効率の問題であって、必ずしも汎用プロセッサ 3 1 - 1 又は 3 1 - 2 で OS を実行する必要はなく、トランザクション処理プロセッサ 3 2 - 1 又は 3 2 - 2 で OS を実行するようにしてもよく、そのような構成を排除するものではない。

## 【 0 0 7 7 】

なお上記実施例の構成においては、汎用プロセッサ 3 1 - 1 及び 3 1 - 2 とトランザクション処理プロセッサ 3 2 - 1 及び 3 2 - 2 との間でメモリ 3 4 を共有することにより、プロセスの移行を効率よく行なうことができる。

## 【 0 0 7 8 】

次に、命令列をプロセッサに割り当てるプロセス割当方法について説明する。

## 【 0 0 7 9 】

図 9 は、本発明の第 1 の実施例によるプロセス割当処理の動作説明図である。

## 【 0 0 8 0 】

プロセスキュー 7 1 に蓄えられるプロセス（或いはスレッド）の割り当ては、OS により管理される。この OS のプロセス割り当ての機能は、ディスパッチャ

72によって実現される。

【0081】

プロセスキュー71は、実行すべきプロセスを格納する。ディスパッチャ72は、プロセスキュー71が示すプロセスと、汎用プロセッサ31-1及び31-2と、トランザクション処理プロセッサ32-1及び32-2とを監視し、プロセスを何れかのプロセッサに割り当てる。本システムは、トランザクション処理が主たる処理である環境で稼動しているので、好ましくは、全てのプロセスは最初に汎用プロセッサではなく、トランザクション処理プロセッサに割り当てられる。

【0082】

或いは、各プロセスを最初に汎用プロセッサではなくトランザクション処理プロセッサに択一的に割り当てるのではなく、例えばトランザクション処理はトランザクション処理プロセッサに割り当てて、トランザクション処理でないことが最初から判明している処理は最初から汎用プロセッサに割り当てるようにしてもよい。

【0083】

システムコールトラップ73は、汎用プロセッサ31-1、31-2、トランザクション処理プロセッサ32-1、32-2からシステムコール（OS割り込み）がかかると、割り込み内容に応じてプロセスの割当を制御する。

【0084】

図10は本発明の第1の実施例によるプロセス割当制御部の処理フローチャートを示す。

【0085】

システムコールトラップ73は、ステップS4-1～S4-5を実行する。

【0086】

ステップS4-1は、汎用プロセッサ31-1、31-2、トランザクション処理プロセッサ32-1、32-2からの割り込みを検出する。ステップS4-1でプロセッサからの割り込みが通知されると、ステップS4-2が実行される。

## 【 0 0 8 7 】

ステップ S 4 - 2 は、割り込みを通知してきたプロセスの命令列を、それまでこのプロセスを実行していたのと同じのプロセッサで、割り込みのかかった命令から継続して処理可能か否かを判定する処理である。ステップ S 4 - 2 で、命令が継続して実行可能であると判断されると、次にステップ S 4 - 3 が実行される。ステップ S 4 - 3 は、プロセスに元のプロセッサを識別する情報を付与する処理である。

## 【 0 0 8 8 】

ステップ S 4 - 3 で元のプロセッサを識別する情報が付与されると、ステップ S 4 - 4 が実行される。ステップ S 4 - 4 は、情報が付与されたプロセスをプロセスキュー 7 1 に格納する処理である。この場合のように、割り込み後に元のプロセッサでプロセスの実行が継続されるのは、例えば I / O 処理に関して割り込みが発生した場合等である。但し、それまでこのプロセスを実行していたプロセッサが、既に別のプロセスに使用されている場合には、他のプロセッサに割り当てを行なう。

## 【 0 0 8 9 】

また、ステップ S 4 - 3 で、プロセスが継続して実行できないと判断された場合には、次にステップ S 4 - 5 が実行される。

## 【 0 0 9 0 】

ステップ S 4 - 5 は、プロセスを実行するプロセッサを汎用プロセッサ或いはトランザクション処理プロセッサに制限する情報を付与する処理を行なう。次に、ステップ S 4 - 4 が実行される。

## 【 0 0 9 1 】

以上により、ディスパッチャ 7 2 がプロセスを割り当てるとき、使用プロセッサに制限が課されている場合には、指定されたプロセッサに当該プロセスを割り当てる。例えば浮動小数点演算を含むプロセスが当初はトランザクション処理プロセッサに割り当てられると、処理過程で割り込みが発生し、汎用プロセッサを指定する情報と共にプロセスがプロセスキュー 7 1 に格納される。これを受けて、ディスパッチャ 7 2 は、このプロセスを汎用プロセッサ 3 1 - 1 又は 3 1 - 2

に割り当てる。

【0092】

なお前述のように、トランザクション処理プロセッサ32-1或いは32-2に全てのプロセスを最初に割り当てることにより、効率的に処理全体を実行することが出来る。これは本システムの処理の主流がトランザクション処理であるためである。或いは、各プロセスを最初に汎用プロセッサではなくトランザクション処理プロセッサに択一的に割り当てるのではなく、例えばトランザクション処理はトランザクション処理プロセッサに割り当てて、トランザクション処理でないことが最初から判明している処理は最初から汎用プロセッサに割り当てるようにしてもよい。

【0093】

またプロセスの割り付けは、プロセッサの性能に応じて最適化することが望ましい。プロセッサの性能評価には、プロセッサに内蔵の通常の性能測定機能を用いればよい。

【0094】

本実施例において、命令セットは、汎用プロセッサ31-1、31-2及びトランザクション処理プロセッサ32-1、32-2で同一であってよい。

【0095】

なお上記実施例では、図4に示すように汎用プロセッサ31-1、31-1及びトランザクション処理プロセッサ32-1、32-2でシステムバス37を共用し、メインメモリ34を共有するシステムについて説明したが、クラスタ方式のマルチプロセッサシステムに本発明を適用することも可能である。

【0096】

図11は本発明の第2の実施例によるシステム構成図を示す。

【0097】

本実施例の情報処理装置51は、汎用プロセッサ部52、トランザクション処理プロセッサ部53、入出力インタフェース54から構成される。

【0098】

汎用プロセッサ部52は、整数演算及び論理演算並びに浮動小数点演算が実行

可能とされていると共に、分岐方向が確定する前に分岐命令を実行するために分岐予測をしたり、オペランド待ちのためのリザベーションを行ったり、オペランドの処理順序変更のためにリオーダ処理を実行すること等が可能となっている。トランザクション処理プロセッサ部 5 3 は、整数演算及び論理演算以外の複雑な演算機能や制御機能は含んでいなくてよく、複数のプロセッサ及び／又は複数のプログラムカウンタを備えるなどして、トランザクション処理を効率的に実行可能な構成とされている。入出力インタフェース 5 4 は、汎用プロセッサ部 5 2 と外部とのインタフェースを提供する。

## 【 0 0 9 9 】

汎用プロセッサ部 5 2 は、2 つの汎用プロセッサ 5 5 - 1、5 5 - 2、メモリ 5 6、システムバス 5 7、システムコントローラ 5 8、クラスタインタフェース 5 9 から構成される。汎用プロセッサ 5 5 - 1、5 5 - 2 は、例えば、図 1 に示すような構成とされている。

## 【 0 1 0 0 】

メモリ 5 6 は RAM から構成され、汎用プロセッサ 5 5 - 1 及び 5 5 - 2 とシステムバス 5 7 を介して接続される。汎用プロセッサ 5 5 - 1 と汎用プロセッサ 5 5 - 2 とで、メモリ 5 6 のメモリ空間が共有される。

## 【 0 1 0 1 】

システムコントローラ 5 8 は、汎用プロセッサ 5 5 - 1 及び 5 5 - 2 からの要求に応じて、システムバス 5 7 及びメモリ 5 6 を制御する。クラスタインタフェース 5 9 は、トランザクション処理プロセッサ部 5 3 とのインタフェースを提供する。

## 【 0 1 0 2 】

また、トランザクション処理プロセッサ 5 3 は、2 つのトランザクション処理プロセッサ 6 0 - 1、6 0 - 2、メモリ 6 1、システムバス 6 2、システムコントローラ 6 3、及びクラスタインタフェース 6 4 を含む。

## 【 0 1 0 3 】

トランザクション処理プロセッサ 6 0 - 1 及び 6 0 - 2 は、図 3 に示すトランザクション処理プロセッサ 1 8 - x と同様な構成でよい。

## 【0104】

メモリ61は、システムバス62を介してトランザクション処理プロセッサ60-1及び60-2と接続される。トランザクション処理プロセッサ60-1及び60-2は、メモリ61のメモリ空間を共有する。

## 【0105】

システムコントローラ63は、トランザクション処理プロセッサ60-1及び60-2からの要求に応じて、システムバス62及びメモリ61を制御する。クラスインタフェース64は、汎用プロセッサ部52のクラスインタフェース59と接続されており、汎用プロセッサ部52とのインタフェースを提供する。

## 【0106】

本実施例では、汎用プロセッサ部52及びトランザクション処理プロセッサ部53は、夫々にメモリ56及びメモリ61を備えており、独立したメモリ空間を有する。汎用プロセッサ部52とトランザクション処理プロセッサ部53との間では、クラスインタフェース59及び64を介して通信が行なわれる。トランザクション処理プロセッサ部53で、例えば浮動小数点演算が必要な命令が発行された場合には、この命令は、クラスインタフェース64及び59を介して汎用プロセッサ部52に供給され、汎用プロセッサ部52により実行される。

## 【0107】

以上、本発明を実施例に基づいて説明したが、本発明は上記実施例に限定されるものではなく、特許請求の範囲に記載の範囲内で様々な変形が可能である。

## 【0108】

## 【発明の効果】

本発明によれば、オールラウンドな高性能を重視して設計されると共に、1つのプログラムの処理を高速に実行することを重視して設計された汎用高性能プロセッサと、小規模な処理を大量に並列実行することを重視して設計されるマルチスプレッド及び／或いはマルチプロセストランザクション処理プロセッサとを設け、トランザクション処理はトランザクション処理プロセッサに割り当てて、トランザクション処理プロセッサが効率的に実行できない処理は汎用高性能プロセッサに割り当てることで、多様な命令を効率よく処理できる情報処理装置を提供



することが出来る。

【図面の簡単な説明】

【図 1】

高性能汎用プロセッサの一例のブロック構成図である。

【図 2】

マルチスプレッドトランザクション処理システムのブロック構成図である。

【図 3】

トランザクション処理プロセッサの一例のブロック構成図である。

【図 4】

本発明の一実施例のシステム構成図である。

【図 5】

本発明の一実施例のトランザクション処理プロセッサのブロック構成図である。

【図 6】

本発明の一実施例のトランザクション処理プロセッサの処理フローチャートである。

【図 7】

本発明の一実施例のトランザクション処理プロセッサの割り込み処理の処理フローチャートである。

【図 8】

本発明の一実施例の汎用プロセッサの処理フローチャートである。

【図 9】

本発明の一実施例の命令割り当て処理の動作説明図である。

【図 1 0】

本発明の一実施例の命令割当制御部の処理フローチャートである。

【図 1 1】

本発明の他の実施例のシステム構成図である。

【符号の説明】

3 1 情報処理システム

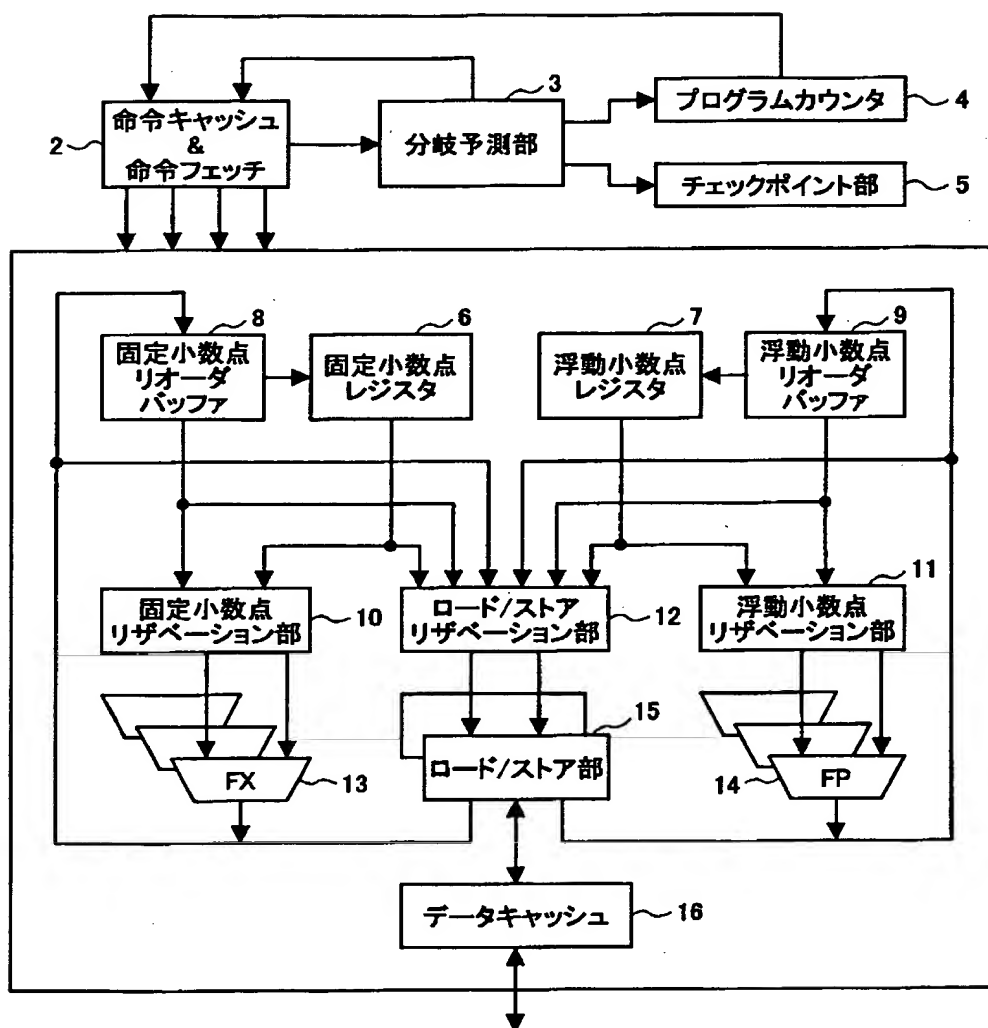
- 3 2 - 1、3 2 - 2 汎用プロセッサ
- 3 3 - 1、3 3 - 2 トランザクション処理プロセッサ
- 3 4 メインメモリ
- 3 5 入出力インタフェース
- 3 6 システムコントローラ
- 3 7 システムバス
- 4 1 トランザクション処理演算部
- 4 2 2次キャッシュメモリ
- 4 3 - 1 ~ 4 3 - m プロセッサコア
- 4 4 2次キャッシュタグ
- 4 5 インタフェース

【書類名】

図面

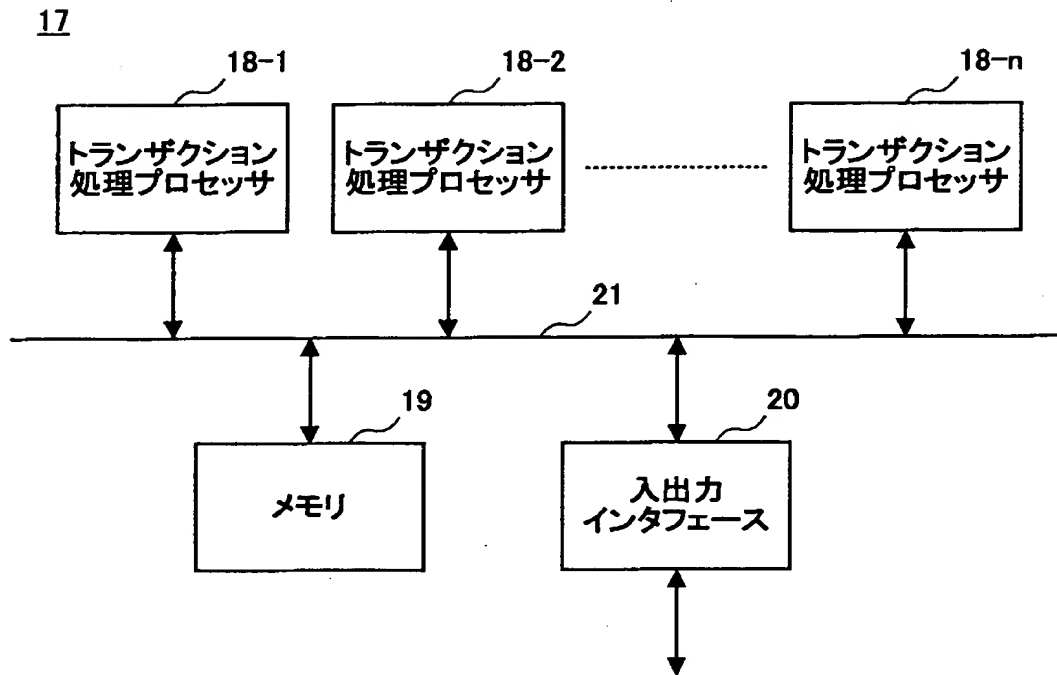
【図 1】

高性能汎用プロセッサの一例のブロック構成図



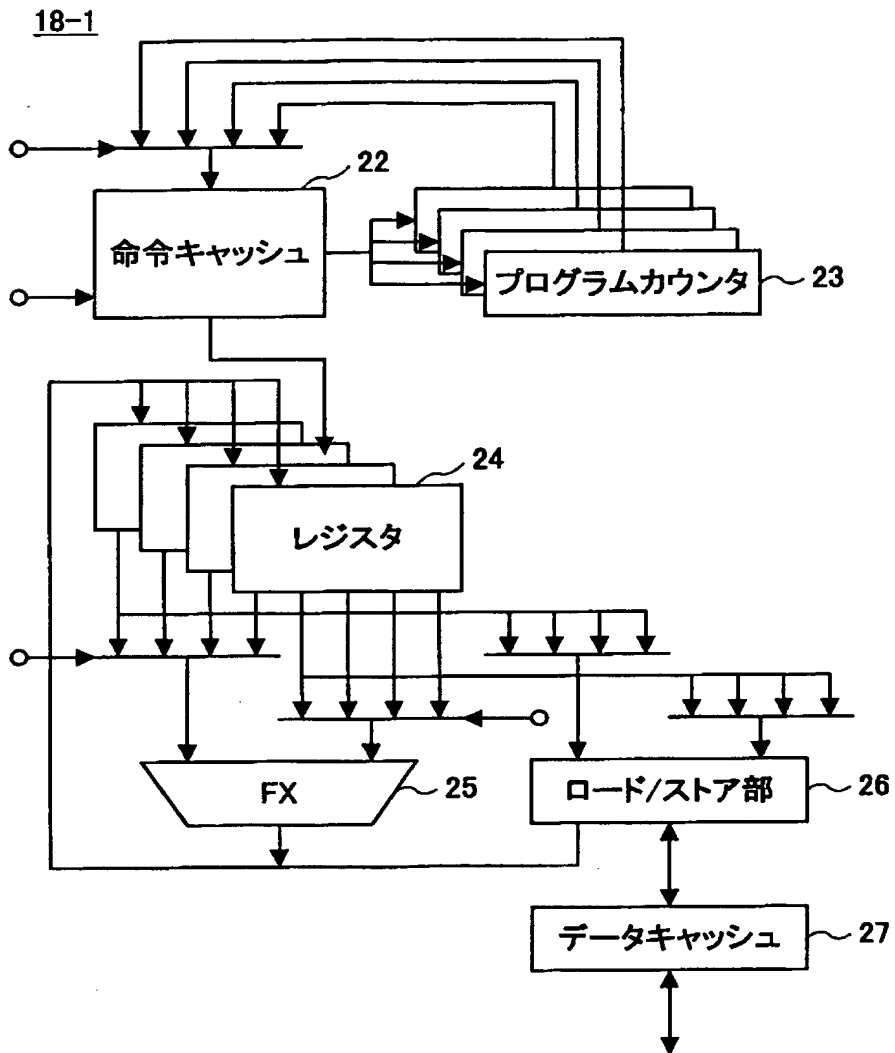
【図 2】

マルチスプレッドトランザクション処理システムの  
ブロック構成図



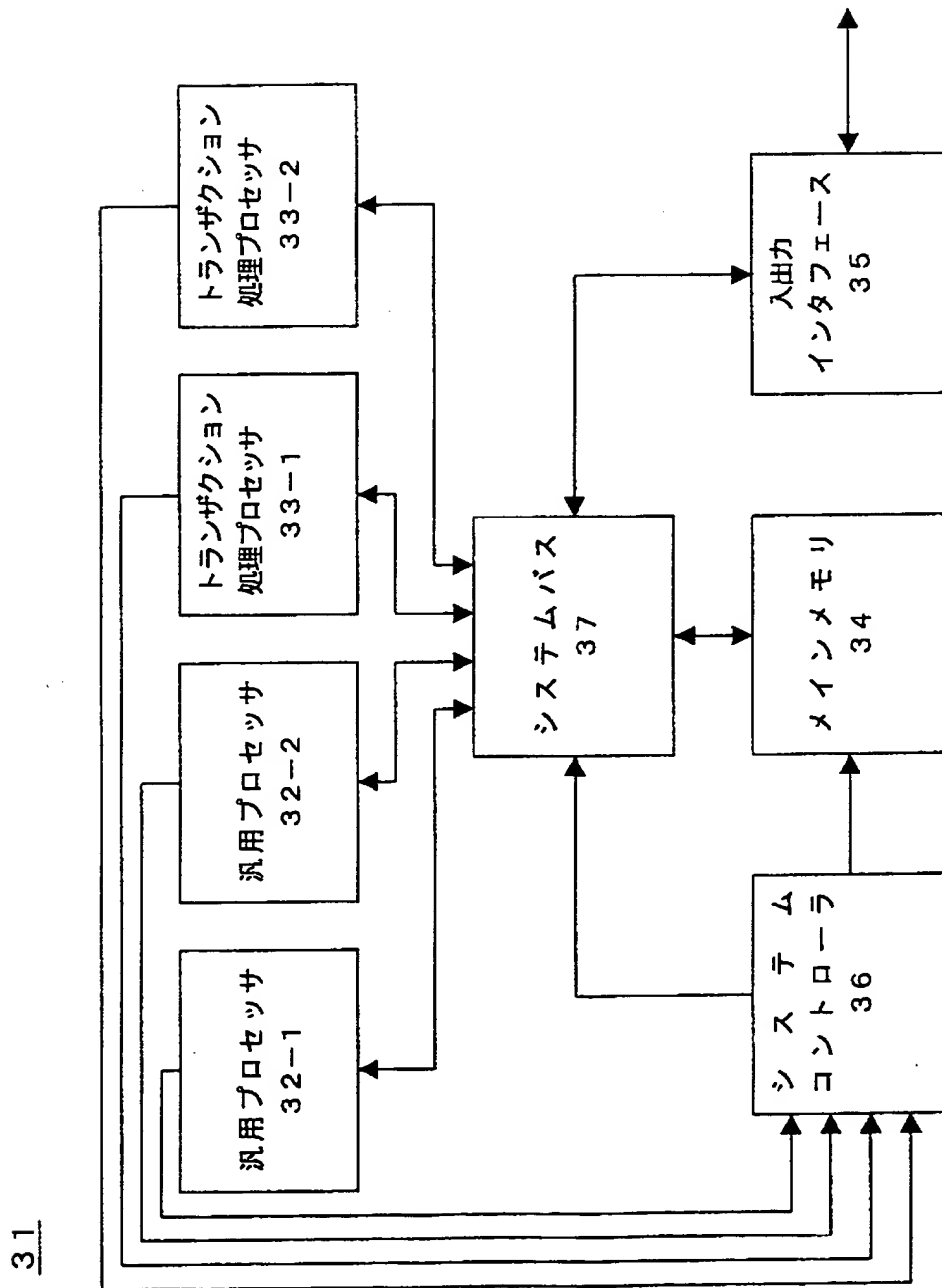
【図 3】

トランザクション処理プロセッサの一例のブロック構成図



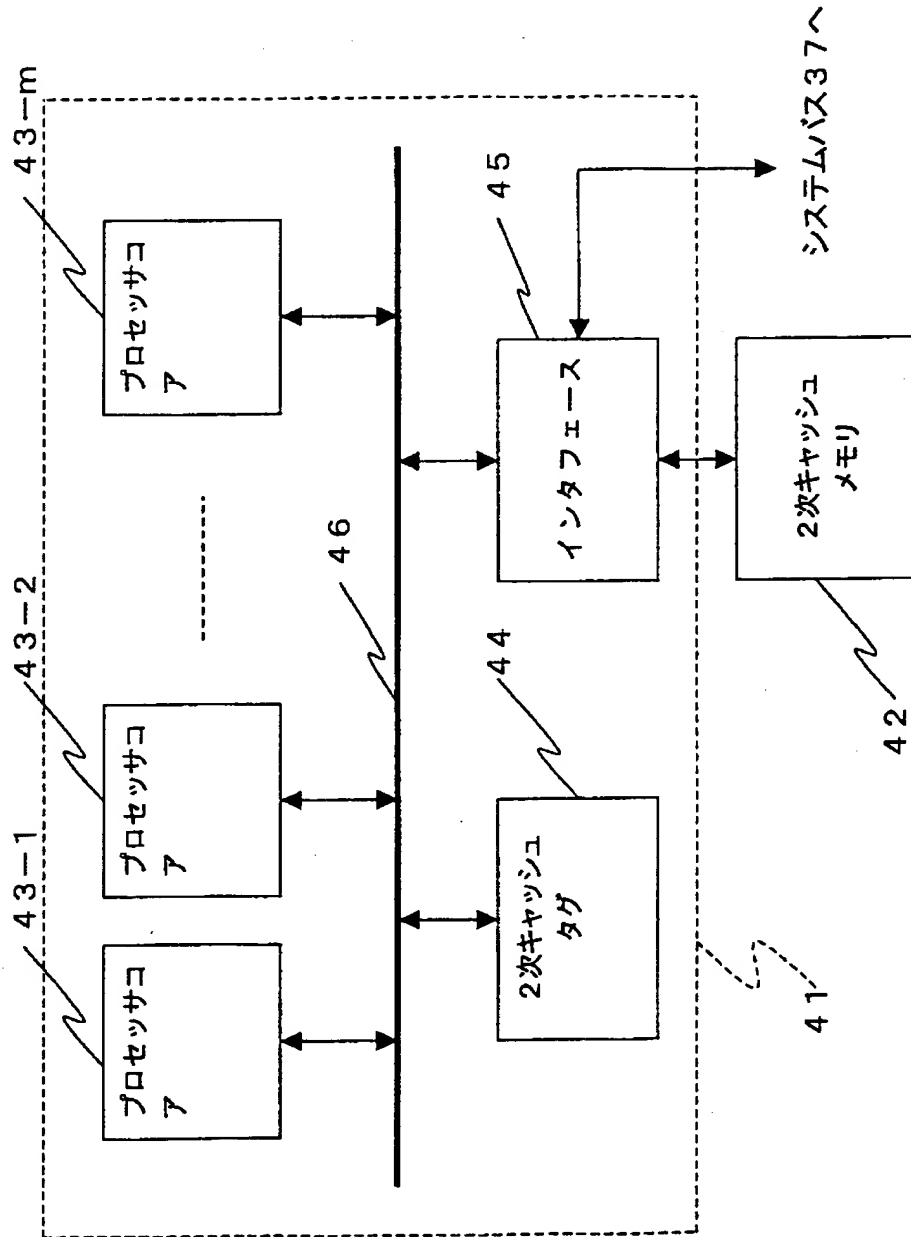
【図 4】

本発明の一実施例のシステム構成図



【図 5】

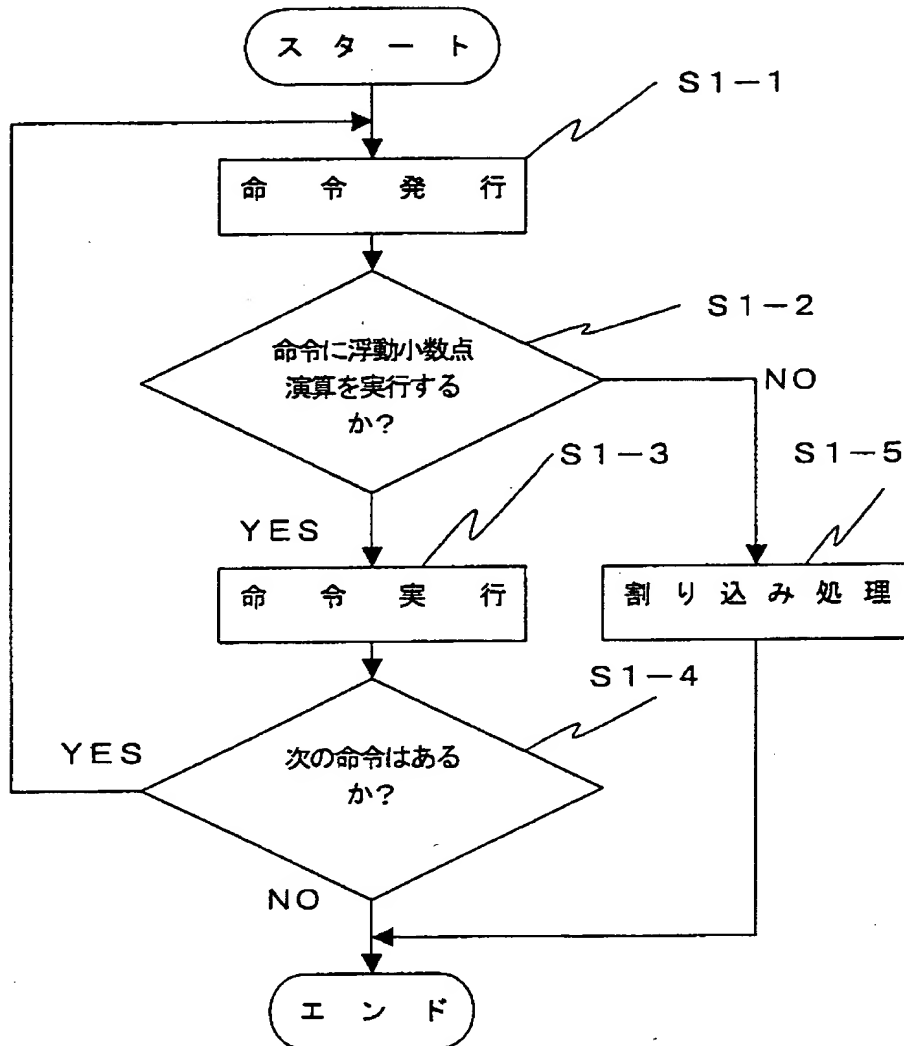
本発明の一実施例のトランザクション処理プロセッサのブロック構成図



33-1、33-2

【図 6】

本発明の一実施例のトランザクション処理プロセッサの処理フローチャート

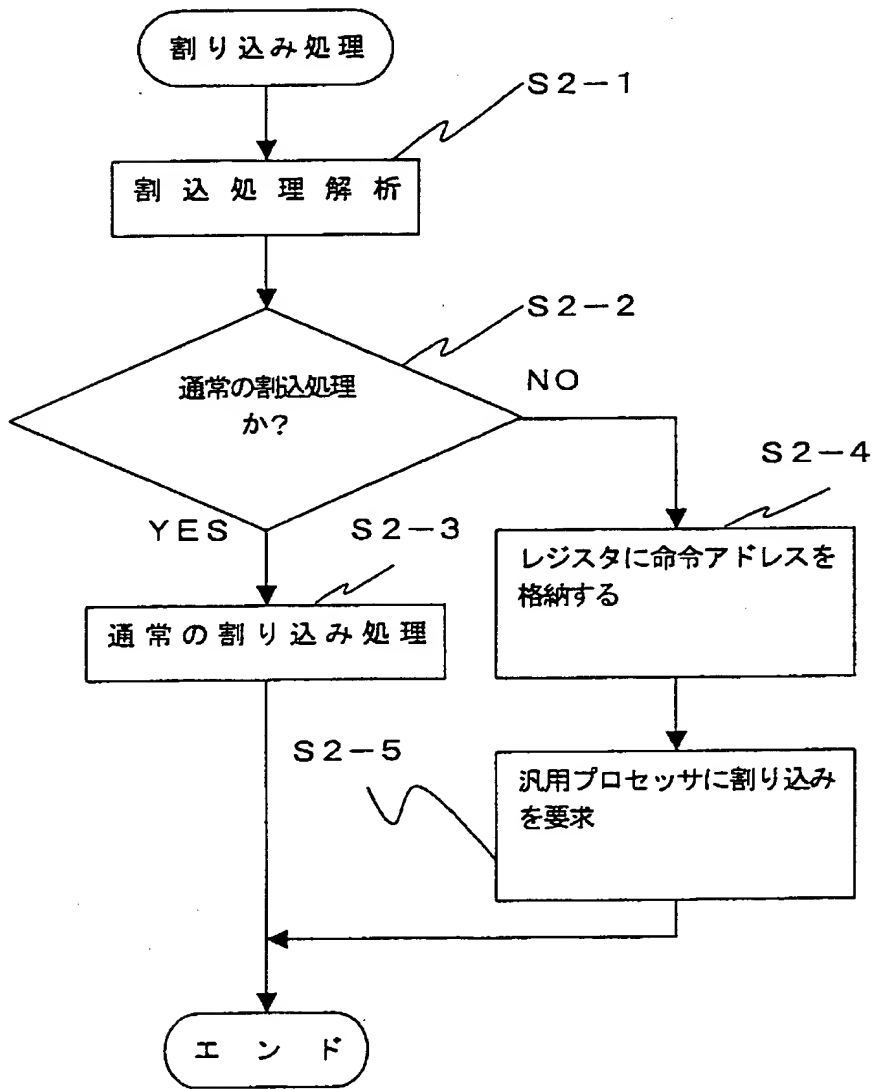




【図 7】

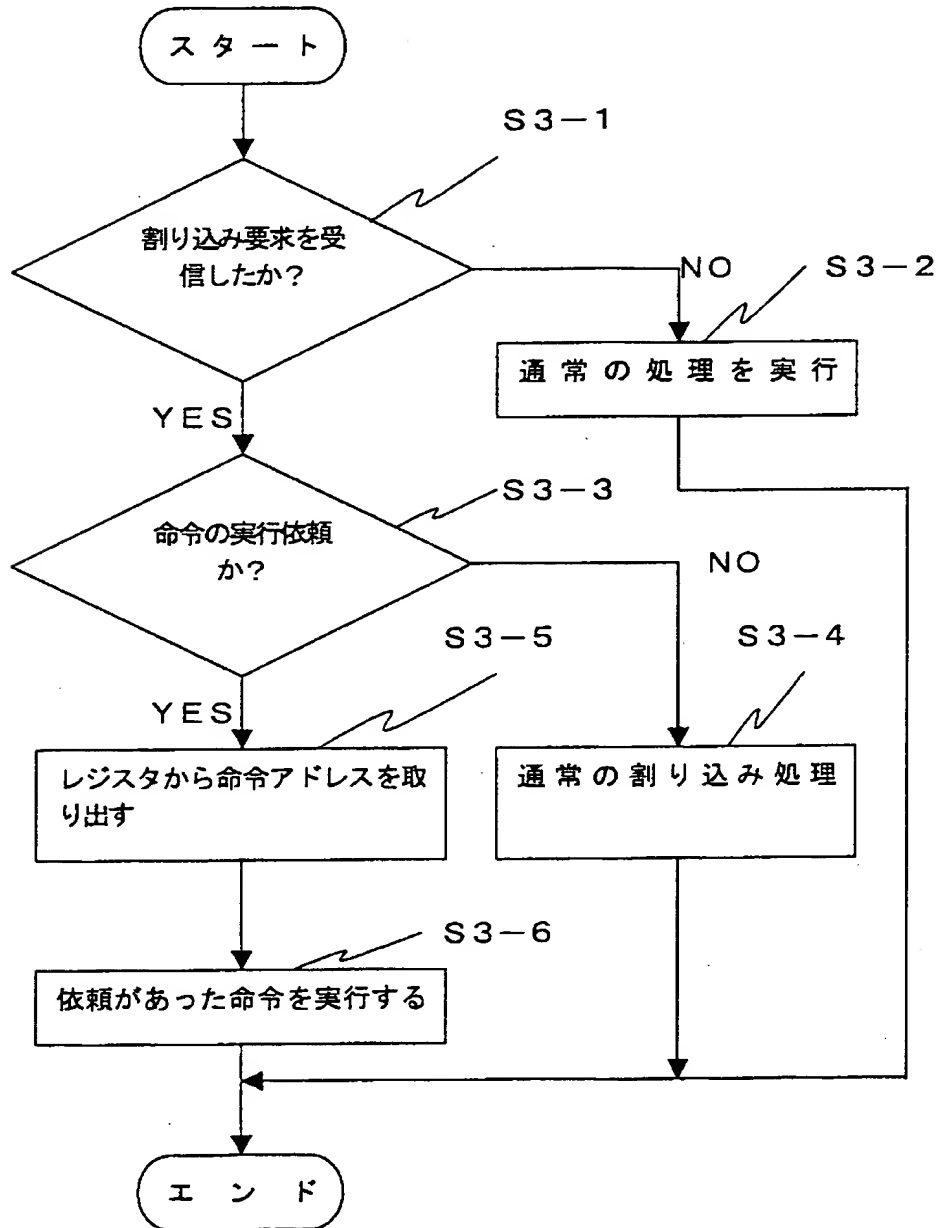
本発明の一実施例のトランザクション処理プロセッサの

割り込み処理フローチャート



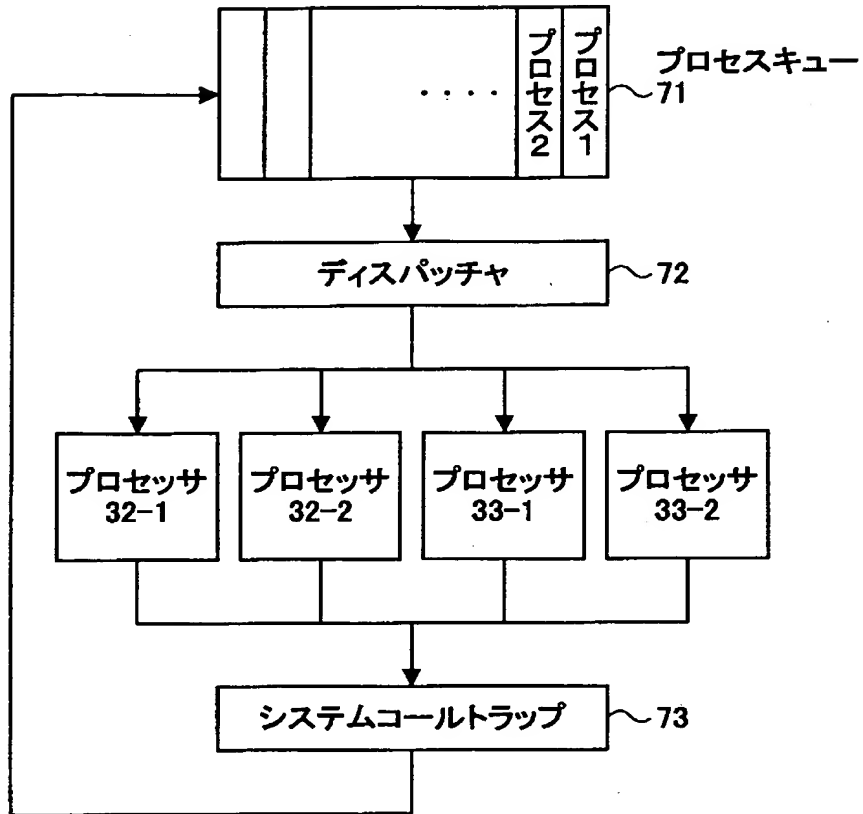
【図 8】

本発明の一実施例の汎用プロセッサの処理フローチャート



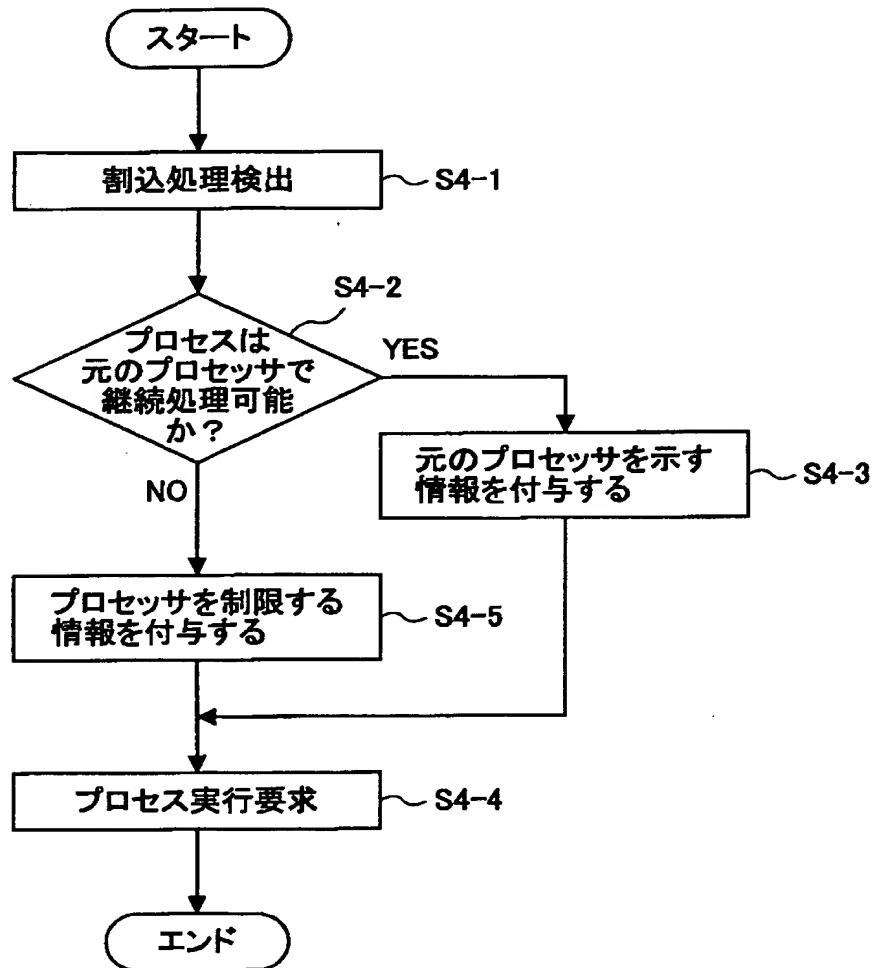
【図 9】

本発明の一実施例の命令割り当て処理の動作説明図



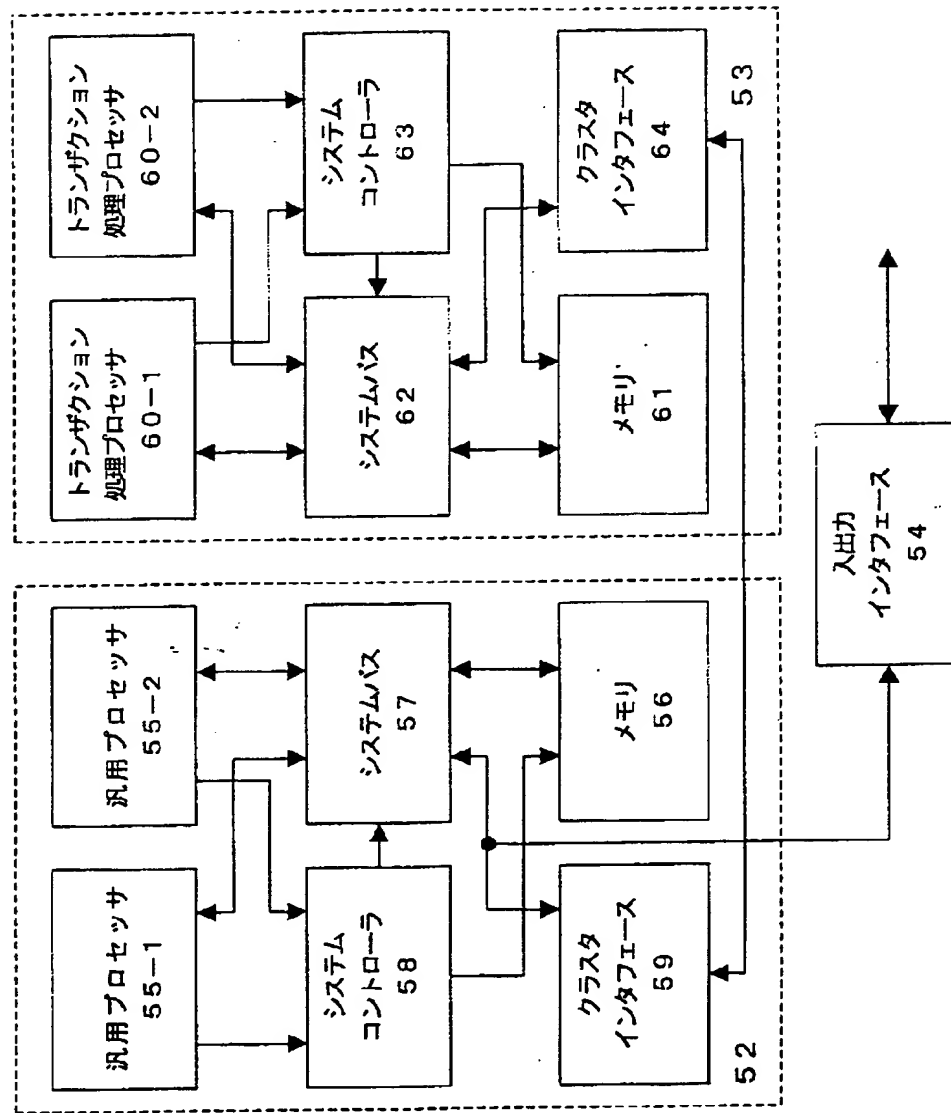
【図 10】

本発明の一実施例の命令割当制御部の処理フローチャート



【図 11】

本発明の一実施例の変形例のブロック構成図



51

【書類名】 要約書

【要約】

【課題】 本発明は、多様な命令を効率よく処理できる情報処理装置を提供することを目的とする。

【解決手段】 種々の処理のうちで特定の処理をそれ以外の処理よりも頻繁に実行する情報処理装置は、種々の処理に対応する命令セットを実行可能な第1のプロセッサと、該命令セットの一部或いは全てを実行可能であり特定の処理に対応する該命令セットの一部を第1のプロセッサより効率的に実行可能な第2のプロセッサを含み、特定の処理を第2のプロセッサが実行し特定の処理以外の処理を第1のプロセッサが実行する。

【選択図】 図4

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社